

EXHIBIT B



US006151606A

United States Patent [19][11] **Patent Number:** **6,151,606****Mendez**[45] **Date of Patent:** **Nov. 21, 2000**

[54] **SYSTEM AND METHOD FOR USING A
WORKSPACE DATA MANAGER TO ACCESS,
MANIPULATE AND SYNCHRONIZE
NETWORK DATA**

5,701,423 12/1997 Crozier 395/335
5,706,502 1/1998 Foley et al. 707/10

(List continued on next page.)

[75] Inventor: **Daniel J. Mendez**, Menlo Park, Calif.

OTHER PUBLICATIONS

[73] Assignee: **Visto Corporation**, Mountain View,
Calif.

Article by Bellovin et al., entitled: "Network Firewalls"
Published by IEEE Communications Magazine Sep. 1994,
pp. 50-57.

[21] Appl. No.: **09/008,354**

Article by Steffen Stempel, entitled: "IPAccess—An Internet
Service Access System for Firewall Installations" Pub-
lished by IEEE Communications Magazine Feb. 16, 1995,
pp. 31-41.

[22] Filed: **Jan. 16, 1998**

[51] **Int. Cl.**⁷ **G06F 17/30**

(List continued on next page.)

[52] **U.S. Cl.** **707/201; 707/8; 707/10;
707/202; 707/203**

[58] **Field of Search** **707/8, 10, 202,
707/203, 506, 511; 709/103, 201, 204,
224, 228, 302, 303; 706/14, 45; 345/302,
340; 320/257, 463; 705/35; 395/500.32**

Primary Examiner—Thomas G. Black

Assistant Examiner—Diane D. Mizrahi

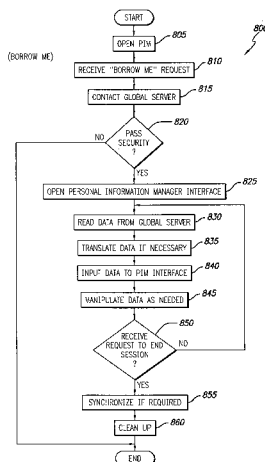
Attorney, Agent, or Firm—Graham & James LLP

[56] **References Cited**

ABSTRACT**U.S. PATENT DOCUMENTS**

4,831,582	5/1989	Miller et al.	707/104
4,875,159	10/1989	Cary et al.	364/200
4,897,781	1/1990	Chang 364/200	
5,263,157	11/1993	Janis 707/9	
5,386,564	1/1995	Shearer et al.	707/500
5,392,390	2/1995	Crozier 395/161	
5,572,643	11/1996	Judson 395/793	
5,581,749	12/1996	Hossain et al.	707/1
5,600,834	2/1997	Howard 395/617	
5,613,012	3/1997	Hoffman et al.	382/115
5,623,601	4/1997	Vu 395/187.01	
5,627,658	5/1997	Connors et al.	358/407
5,634,053	5/1997	Noble et al. 707/4	
5,647,002	7/1997	Brunson 380/49	
5,652,884	7/1997	Palevich 395/651	
5,666,530	9/1997	Clark et al. 395/617	
5,666,553	9/1997	Crozier 395/803	
5,678,039	10/1997	Hinks et al. 395/604	
5,680,542	10/1997	Mulchandani 395/183.04	
5,682,524	10/1997	Freund et al. 395/605	
5,684,990	11/1997	Boothby 707/203	
5,687,322	11/1997	Deaton 705/14	
5,701,400	12/1997	Amado 706/45	

A system includes a communications module for download-
ing workspace data from a remote site, an application
program interface coupled to the communications module
for communicating with a workspace data manager to enable
manipulation of the downloaded workspace data and thereby
create manipulated data, and a general synchronization
module coupled to the communications module for synchro-
nizing the manipulated data with the workspace data stored
at the remote site. An instantiator requests the workspace
data manager to provide an interface for enabling manipu-
lation of the downloaded workspace data. The workspace
data manager may create another instance of the interface or
may provide access to its only interface to enable manipu-
lation of the data. A data reader may translate the down-
loaded workspace data from the format used by the remote
site to the format used by the workspace data manager. Upon
logout, a de-instantiator synchronizes the data with the
global server and deletes the workspace data. The system
handles the situation where the data stored at the remote site
has not changed and therefore includes the downloaded data,
and the situation the data stored at the remote site has been
modified and therefore is different than the downloaded data.

21 Claims, 6 Drawing Sheets

6,151,606

Page 2

U.S. PATENT DOCUMENTS

5,710,918	1/1998	Lagarde et al.	395/610
5,713,019	1/1998	Keaten	395/610
5,715,403	2/1998	Stefik	705/44
5,717,925	2/1998	Harper et al.	395/613
5,721,908	2/1998	Lagarde et al.	395/610
5,721,914	2/1998	DeVries	707/104
5,729,735	3/1998	Meyering	395/610
5,745,360	4/1998	Leone et al.	364/140
5,757,916	5/1998	MacDoran et al.	380/25
5,758,150	5/1998	Bell et al.	395/610
5,758,354	5/1998	Huang et al.	707/201
5,758,355	5/1998	Buchanan	707/201
5,765,171	6/1998	Gehani et al.	707/203
5,778,346	7/1998	Frid-Nielsen et al.	395/208
5,790,425	8/1998	Wagle	364/551.01
5,790,790	8/1998	Smith et al.	395/200.36
5,799,318	8/1998	Cardinal et al.	707/104
5,832,483	11/1998	Barker	707/8
5,862,325	1/1999	Reed	395/200.31
5,870,759	2/1999	Bauer et al.	707/201
5,951,652	9/1999	Ingrassia et al.	709/248
5,966,714	10/1999	Huang et al.	707/201
5,999,947	12/1999	Zollinger et al.	707/203

OTHER PUBLICATIONS

Article by Braun et al., entitled: "Web Traffic Characterization: an assessment of the impact of caching documents from NCSA's web server" Published by Elsevier Science B.V. 1995 pp. 37-51.

Article by Nelson et al., entitled: "Security for Infinite Networks" Published by IEEE Communications Magazine on Aug. 22, 1995, pp. 11-19.

Article by Greenwald et al., entitled: "Designing an Academic Firewall: Policy, Practice, and Experience with SURF" Published by IEEE Communications Magazine on Feb. 22, 1996, pp. 79-92.

Article by Kiuchi et al., entitled: "C-HTTP—The Development of a Secure, Closed HTTP-based Network on the Internet" Published by IEEE Proceedings of SNDSS on Feb. 22, 1996, pp. 64-75.

Article by S. Cobb, entitled: "Establishing Firewall Policy" Published by National Computer Security Assn. on Jun. 25-27, 1996, pp. 198-205.

Margaret J. Brown, "The Visto Briefcase Pro Puts Your PIM On The Internet", URL:http://www.zdnet.com/zdnn/stories/zdnn_display/0,3440,341892,00.html, Aug. 13, 1998, 1 page.

Web site entitled "Bookmark Translator 2.0: This Utility transform Microsoft Internet Explore's bookmarks in the format valid for Netscape Navigator and viceversa," Enzo Marinacci, Rome-Jul. 1997, URL=<http://www.bns.it/emware/BookmarkTranslator-uk.htm>, pp. 1-4.

U.S. Patent

Nov. 21, 2000

Sheet 1 of 6

6,151,606

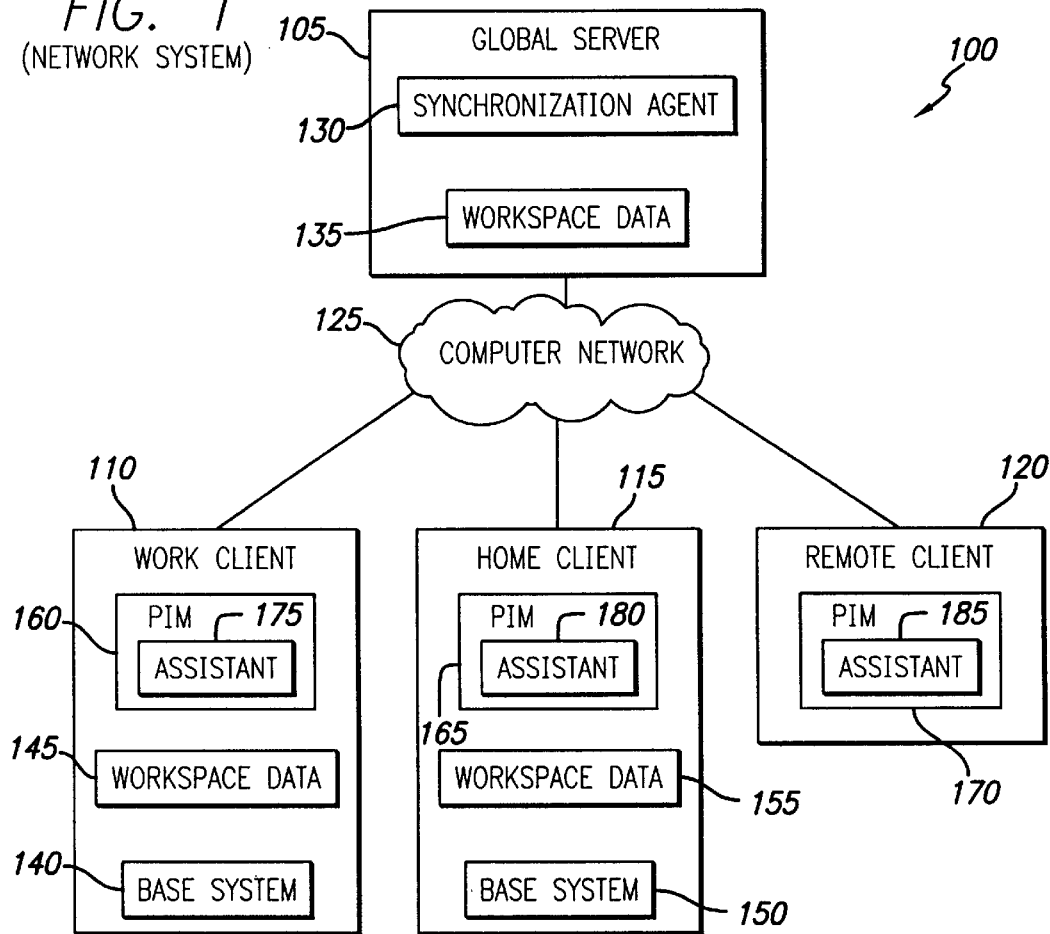
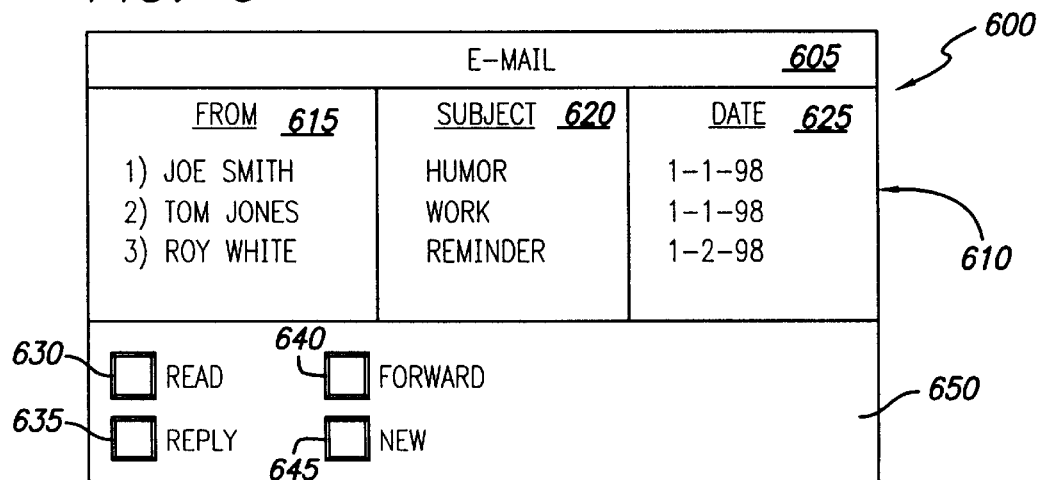
FIG. 1
(NETWORK SYSTEM)

FIG. 6

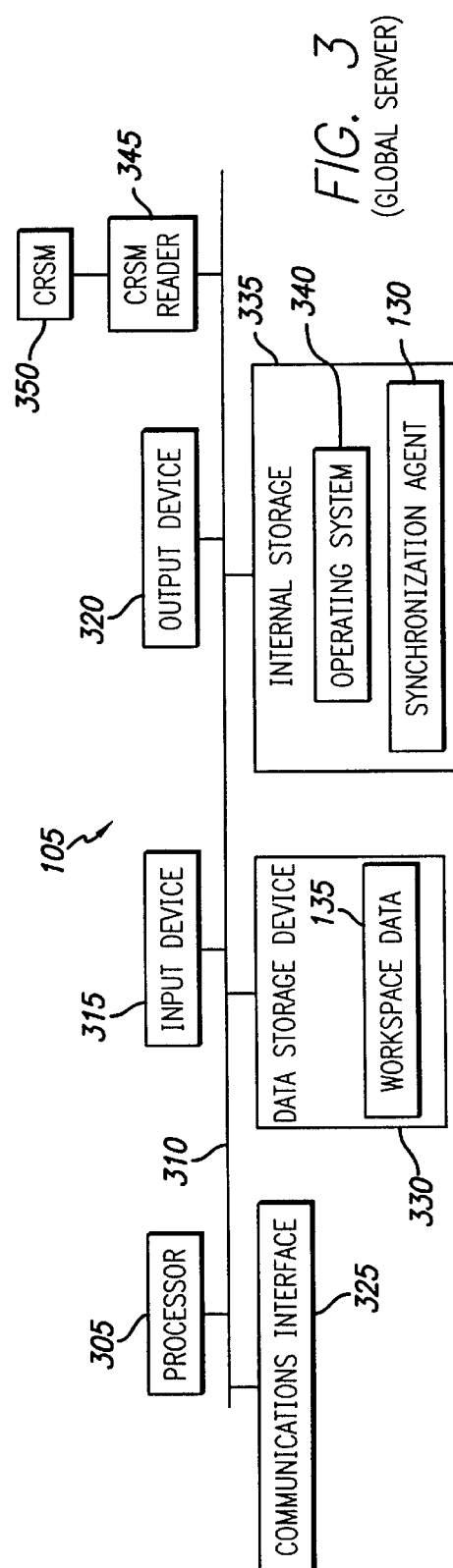
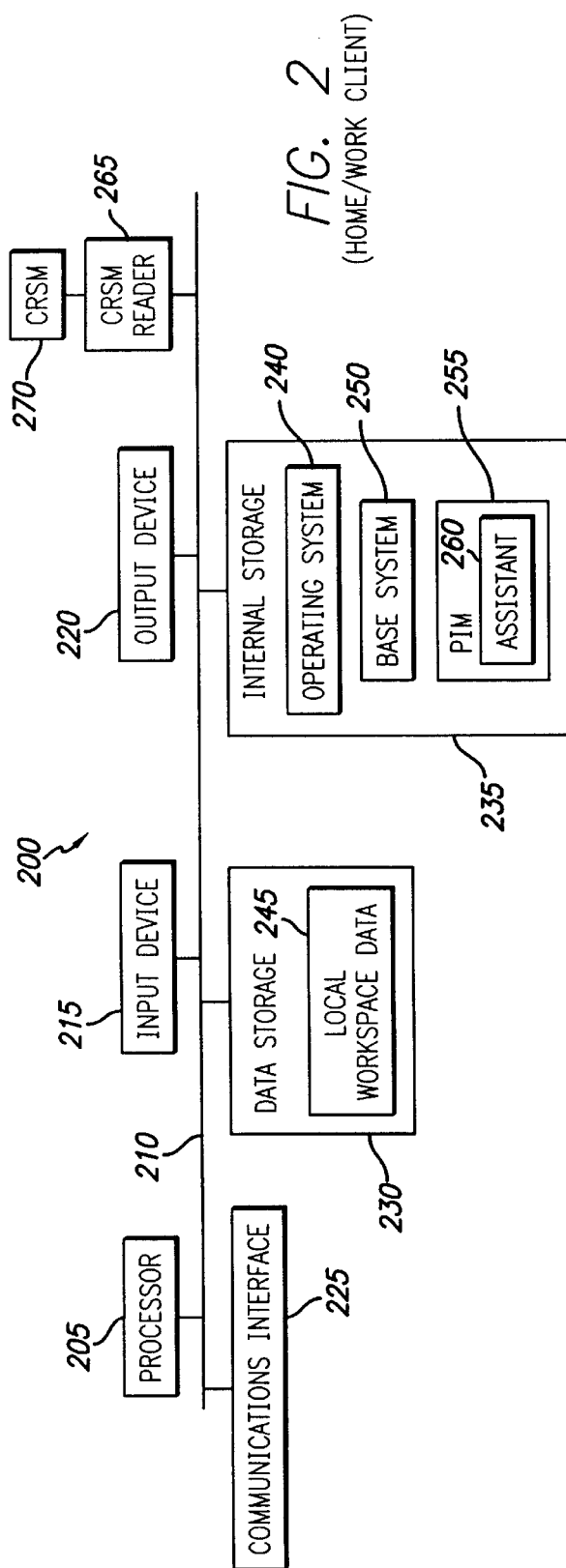


U.S. Patent

Nov. 21, 2000

Sheet 2 of 6

6,151,606

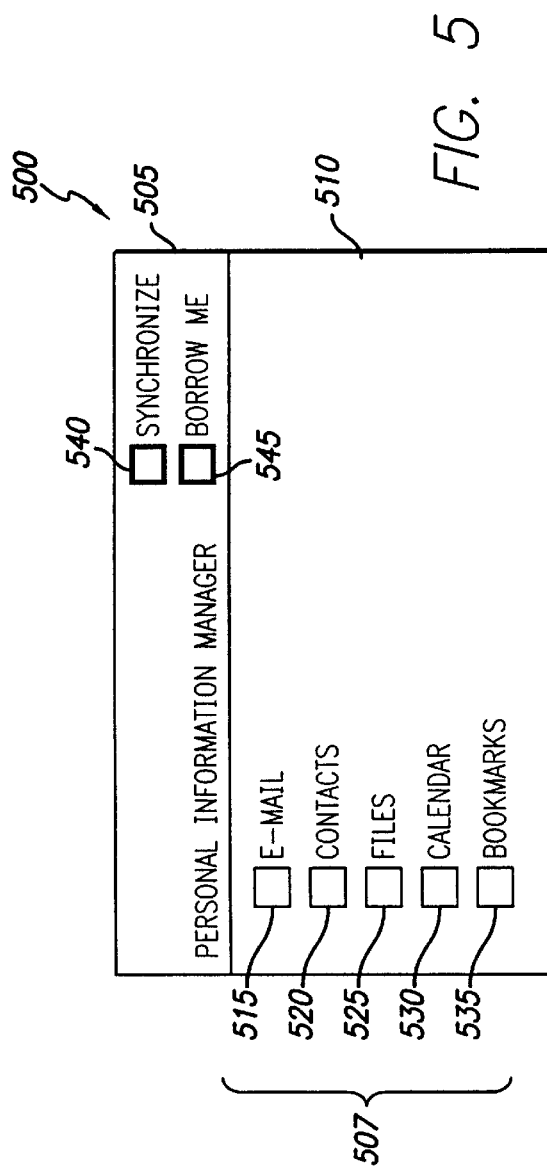
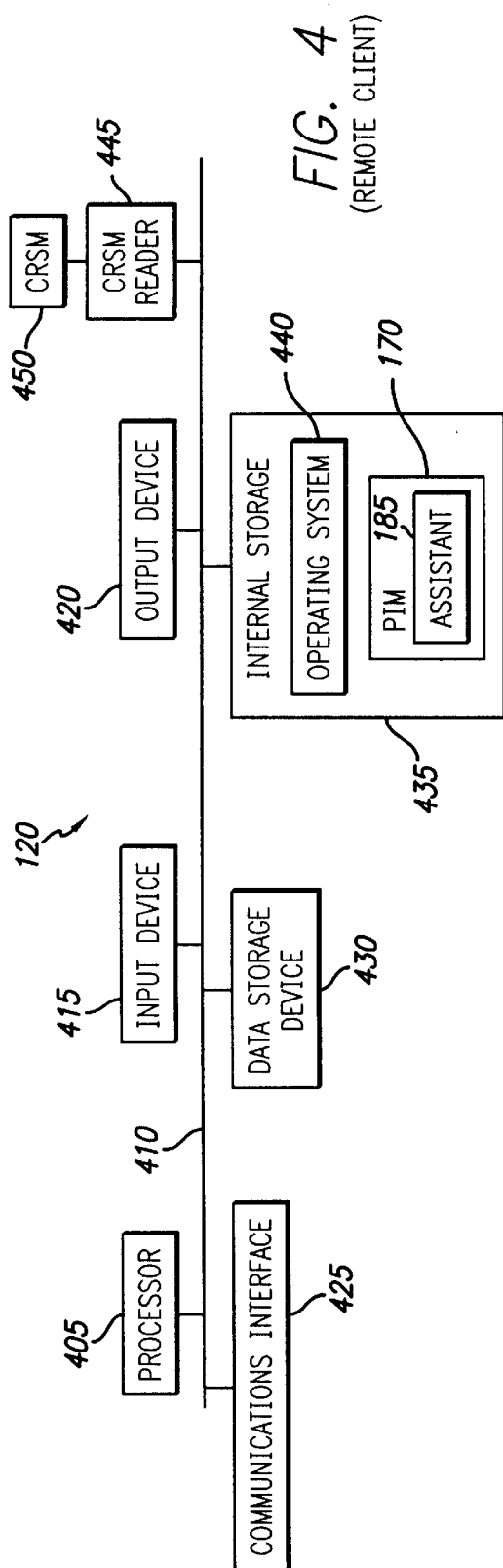


U.S. Patent

Nov. 21, 2000

Sheet 3 of 6

6,151,606

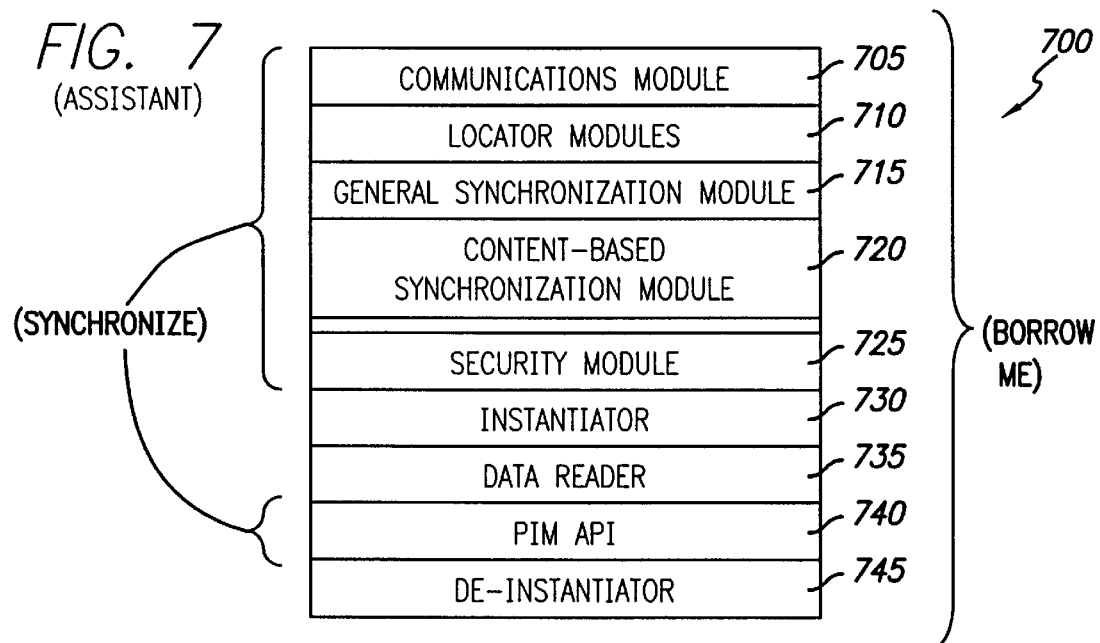


U.S. Patent

Nov. 21, 2000

Sheet 4 of 6

6,151,606



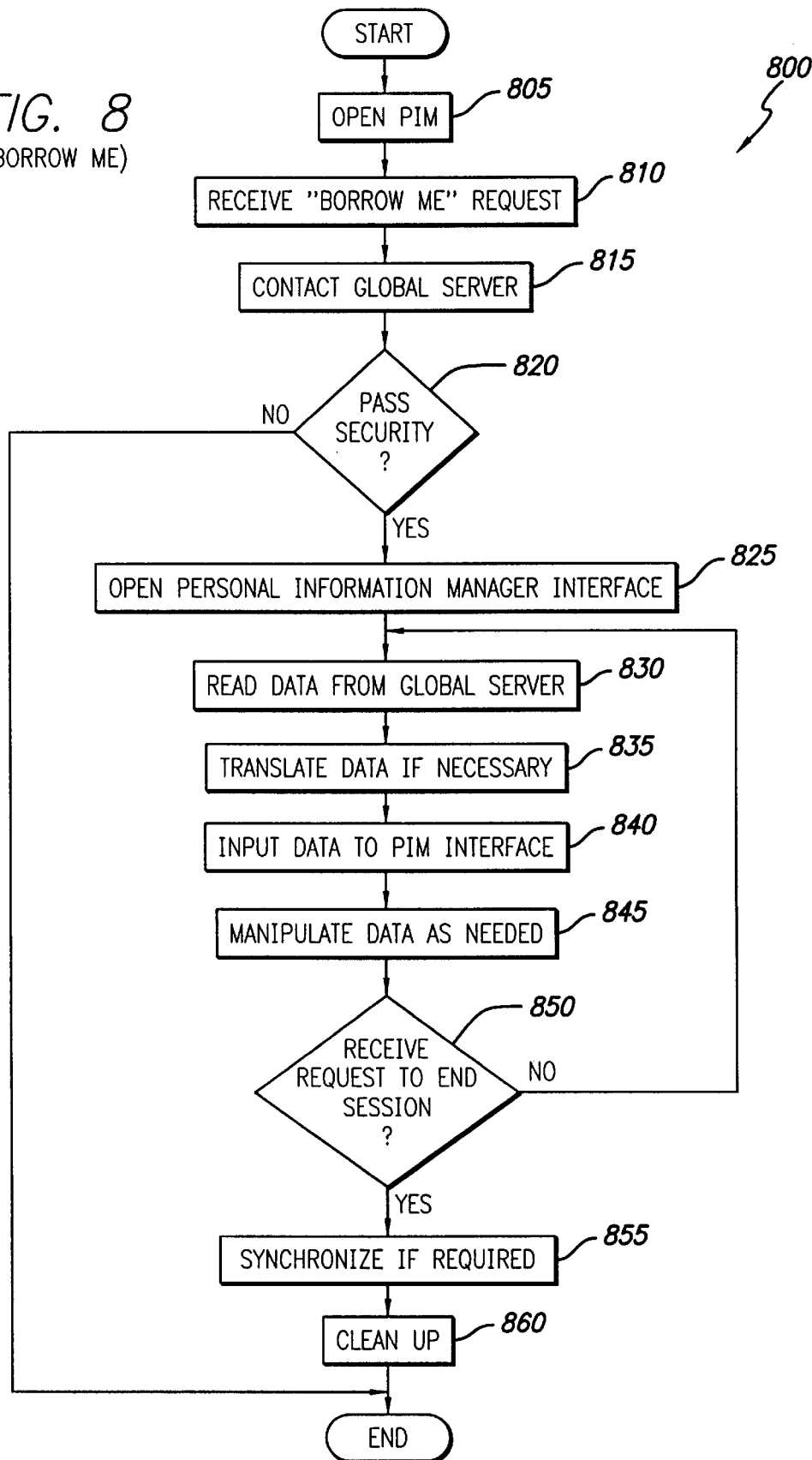
U.S. Patent

Nov. 21, 2000

Sheet 5 of 6

6,151,606

FIG. 8
(BORROW ME)



U.S. Patent

Nov. 21, 2000

Sheet 6 of 6

6,151,606

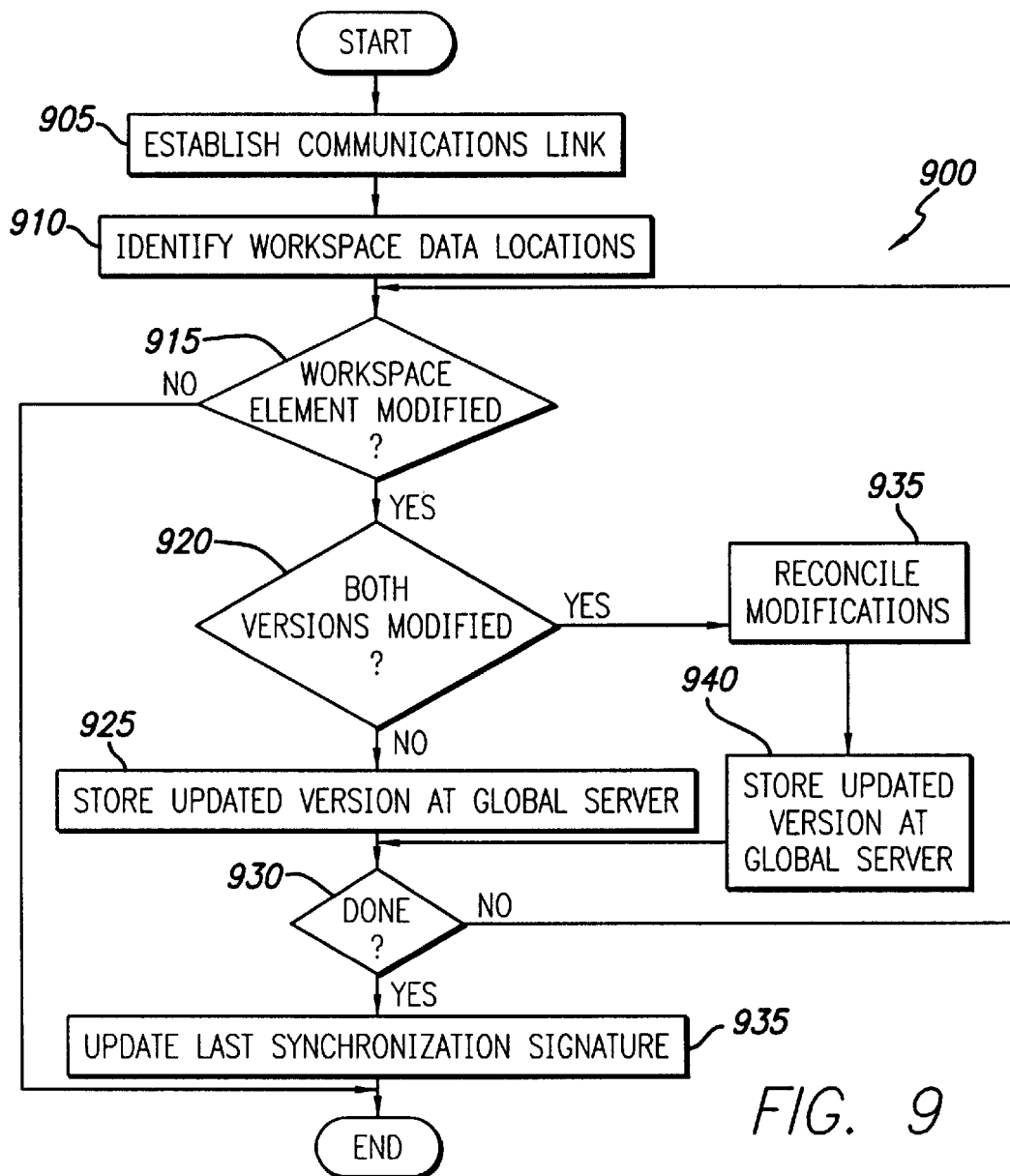


FIG. 9

6,151,606

1

SYSTEM AND METHOD FOR USING A WORKSPACE DATA MANAGER TO ACCESS, MANIPULATE AND SYNCHRONIZE NETWORK DATA

PRIORITY REFERENCES

This application claims priority of and hereby incorporates by reference U.S. patent application Ser. No. 08/766,307 pending, entitled "System and Method for Globally Accessing Computer Services," filed on Dec. 13, 1996, by inventors Mark D. Riggins, et al; U.S. patent application Ser. No. 08/841,950 pending, entitled "System and Method for Enabling Secure Access to Services in a Computer Network," filed on Apr. 8, 1997, by inventor Mark D. Riggins; U.S. patent application Ser. No. 08/865,075, and now U.S. Pat. No. 6,023,708 entitled "System and Method for Using a Global Translator to Synchronize Workspace Elements Across a Network," filed on May 29, 1997, by inventors Daniel J. Mendez, et al.; U.S. patent application Ser. No. 08/835,997 pending, entitled "System and Method for Securely Synchronizing Multiple Copies of a Workspace Element in a Network," filed on Apr. 11, 1997, by inventors Daniel J. Mendez, et al.; U.S. patent application Ser. No. 08/897,888 pending and now U.S. Pat. No. 5,961,590, entitled "System and Method for Synchronizing Electronic Mail Across a Network," filed on Jul. 22, 1997, by inventors Daniel J. Mendez, et al.; U.S. patent application Ser. No. 08/899,277, entitled "System and Method for Using an Authentication Applet to Identify and Authenticate a User in a Computer Network," filed on Jul. 23, 1997, by inventor Mark D. Riggins; and U.S. patent application Ser. No. 08/903,118 pending, entitled "System and Method for Globally and Securely Accessing Unified Information in a Computer Network," filed on Jul. 30, 1997, by inventors Daniel J. Mendez, et al.

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to computer networks, and more particularly provides a system and method for using a workspace data manager to access network data.

2. Description of the Background Art

Data accessibility and consistency are significant concerns for computer users. For example, when a roaming user, i.e., a user who travels to a remote location, needs to review or manipulate data such as an e-mail or prepared document, the roaming user must either carry the data to the remote location or access a workstation remotely. Maintaining a true copy of a database is a cumbersome process. Accordingly, system designers have developed an array of techniques for connecting a remote terminal across a computer network to the workstation storing the data.

To guarantee readability of the downloaded data, the user must carry a laptop computer containing all the applications needed to present and enable manipulation of the downloaded data, or find a network-connected computer that contains the needed application programs. Further, when maintaining multiple independently modifiable copies of particular data, a user risks using an outdated version. By the time the user notices an inconsistency, interparty miscommunication or data loss may already have resulted. The user must then spend more time reconciling the inconsistent versions.

The problems of data accessibility and inconsistency are exacerbated when multiple copies of a document are main-

2

tained at different network locations. For example, due to network security systems such as conventional firewall technology, a user may have access only to a particular one of these network locations. Without access to the other sites, the user cannot confirm that the version on the accessible site is the most recent draft.

SUMMARY OF THE INVENTION

The present invention provides a system for using a workspace data manager to access, manipulate and synchronize workspace data. A workspace data manager may include a Personal Information Manager (PIM), a word processing program, a spreadsheet program, or any application program that enables manipulation of workspace data. Workspace data includes at least one workspace element, such as an e-mail, a day of calendar data, a word document, a bookmark, a sheet of spreadsheet data, or a portion thereof. Workspace data may include e-mails, calendar data, word documents, bookmarks, spreadsheet data, or portions thereof.

The system includes a communications module for downloading workspace data from a remote site, an application program interface coupled to the communications module for communicating with a workspace data manager to enable manipulation of the downloaded workspace data and thereby create manipulated data, and a general synchronization module coupled to the communications module for synchronizing the manipulated data with the workspace data stored at the remote site. An instantiator requests the workspace data manager to provide an interface for enabling manipulation of the downloaded workspace data. The workspace data manager may create another instance of the interface or may provide access to its only interface to enable manipulation of the data. A data reader translates the downloaded workspace data from the format used by the remote site to the format used by the workspace data manager. For example, data stored at the global server site in a canonical format may be translated to Organizer™, Outlook™ or other workspace element manager format. Upon logout, a de-instantiator initiates synchronization and deletes the data stored locally. It will be appreciated that the system handles the situation where the data stored at the remote site has not changed and therefore includes the downloaded data, and the situation the data stored at the remote site has been modified and therefore is different than the downloaded data.

The present invention further provides a method of using a workspace data manager to enable access, manipulate and synchronize workspace data. The method comprises the steps of downloading data from a remote site, requesting a workspace data manager to enable manipulation of the data and thereby create manipulated data, and synchronizing the manipulated data with the data stored at the remote site.

The system and method of the present invention advantageously enable the use of an integral interface, instead of using an interface for the synchronization software, an interface for the workspace data manager and an interface for the communication engine downloading the workspace data. Accordingly, the user need not become familiar with multiple interfaces. The user need only find a remote site that includes a workspace data manager that includes assistant-like functionality. Assistant-like functionality includes services for interfacing between the workspace data manager and the global server. Because the system and method substitute the global data for the local data, or create an instance for the global data, the system and method further advantageously enable a workspace data manager to provide

6,151,606

3

an interface for manipulating workspace data without compromising the local data.

Further, the system and method advantageously provide a simple graphical user interface for enabling borrowing of the workspace data manager and synchronization of manipulated data. The system and method also advantageously delete downloaded data and all interfaces from the local client, so that no traces are left on the local client for unprivileged users to review. Using the technology described in the applications incorporated by reference above, the system and method of the present invention further enable access and synchronization of data across different workspace data manager formats and across network firewalls.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a network system, in accordance with the present invention;

FIG. 2 is a block diagram illustrating details of the home or work client of FIG. 1;

FIG. 3 is a block diagram illustrating details of the global server of FIG. 1;

FIG. 4 is a block diagram illustrating details of the remote client of FIG. 1;

FIG. 5 is a block diagram illustrating details of an assistant of FIG. 1;

FIG. 6 illustrates a personal information manager interface;

FIG. 7 illustrates a second personal information manager interface incorporating an assistant interface;

FIG. 8 is a flowchart illustrating a method of accessing network data from a remote site in accordance with the present invention; and

FIG. 9 is a flowchart illustrating a method of synchronizing network data from a remote site.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 is a block diagram illustrating a network system 100 for using a workspace data manager to access, manipulate and synchronize workspace data in accordance with the present invention. A workspace data manager may include a Personal Information Manager (PIM), a word processing program, a spreadsheet program, or any application program that enables manipulation of workspace data. Workspace data includes at least one workspace element, such as an e-mail, a day of calendar data, a word document, a bookmark, a sheet of spreadsheet data, or a portion thereof. Workspace data may include e-mails, calendar data, word documents, bookmarks, spreadsheet data, or portions thereof. Although the network system 100 is described with reference to PIM's, one skilled in the art will recognize that the system 100 will work with any workspace data manager.

Network system 100 includes a global server 105 coupled via a computer network 125 to a work client 110, to a home client 115 and to a remote client 120. The global server 105 includes a synchronization agent 130 and workspace data 135. The work client 110 includes a base system 140 and workspace data 145. The home client 115 includes a base system 150 and workspace data 155.

Each of the base system 140 and the base system 150 cooperate with the synchronization agent 130 to synchronize workspace data 135, workspace data 145 and workspace data 155 between the work client 110, the home client 115

4

and the global server 105. Synchronization of workspace data 135, 145 and 155 is described in detail in the patent applications incorporated by reference above. However, a brief example of synchronization is provided for completeness.

First, the base system 140 on the work client 110 site negotiates a secure communications channel via any firewalls with the synchronization agent 130, for example, using Secure Sockets Layer (SSL) technology. The base systems 140 examines version information and if necessary the content of a workspace to determine the most updated version. The most updated version is then stored at the client 110 site and at the global server 105 site. The base system 140 repeats these operations for all workspace elements selected for synchronization. Second, the base system 150 on the home client 115 site uses similar steps to synchronize its workspace data 155 with the workspace data 135 on the global server 105 site. Accordingly, the most updated versions of the workspace data 135, 140 and 145 are stored at all three sites.

Each of the work client 110, the home client 115 and the remote client 120 includes a respective workspace data manager, e.g., a Personal Information Manager (PIM) 160, 165 and 170 such as Outlook™ 98 developed by Microsoft Corporation, Organizer 97 developed by Lotus Development Corporation or Sidekick 98 developed by Starfish Software. Each PIM 160, 165 and 170 includes an assistant 175, 180 and 185 that adds data access and synchronization functions to the PIM 160, 165 and 170. Accordingly, a user can transparently use an assistant 175, 180 or 185 via a PIM 160, 165 or 170 to access workspace data 135 from the global server 105, to present and enable manipulation of downloaded workspace data 135, and to synchronize manipulated downloaded data 135 with the workspace data 135 stored on the global server 105. Components and operations of the assistant 175, 180 or 185 are described in detail with reference to FIGS. 7-9.

FIG. 2 is a block diagram illustrating details of a data-synchronizing client 200, in a generic embodiment which exemplifies each of the work client 110 and the home client 115. The client 200 includes a processor 205, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a communications channel 210. The client 200 further includes an input device 215 such as a keyboard and mouse, an output device 220 such as a Cathode Ray Tube (CRT) display, data storage 230 such as a magnetic disk, and internal storage 235 such as Random-Access Memory (RAM), each coupled to the communications channel 210. A communications interface 225 couples the communications channel 210 to the computer network 125.

An operating system 240 controls processing by processor 205, and is typically stored in data storage 230 and loaded into internal storage 235 (as illustrated) for execution. A base system 250, which cooperates with the synchronization agent 130 for synchronizing local workspace data 245 with workspace data 135, also may be stored in data storage 230 and loaded into internal storage 235 (as illustrated) for execution by processor 205. The local workspace data 245 exemplifies workspace data 145 or workspace data 150, and may be stored in data storage 230.

A PIM 255 includes an assistant 260, which enables a user to download workspace data 135 from the global server 105, and to use the PIM 255 for displaying and manipulating the workspace data 135. The assistant 260 further enables the PIM 255 to synchronize the manipulated data 135 with the

6,151,606

5

workspace data **135** on the global server **105**. The PIM **255** exemplifies each of the PIM **160** on the work client **110** and the PIM **165** on the home client **115**. The assistant **260** exemplifies each of the assistant **175** on the work client **110** and the assistant **180** on the home client **115**. The PIM **255** may be stored in data storage **230**, and loaded into internal storage **235** (as illustrated) for execution by the processor **205**.

One skilled in the art will recognize that the system **100** may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system **100** in alternative ways. For example, a computer-readable storage medium (CRSM) reader **265** such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus **210** for reading a computer-readable storage medium (CRSM) **270** such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the system **100** may receive programs and data via the CRSM reader **265**.

FIG. **3** is a block diagram illustrating details of the global server **105**. The global server **105** includes a processor **305**, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a communications channel **310**. The global server **105** further includes an input device **315** such as a keyboard and mouse, an output device **320** such as a CRT display, data storage **325** such as a magnetic disk, and internal storage **330** such as RAM, each coupled to the communications channel **310**. A communications interface **325** couples the communications channel **310** to the computer network **125**.

An operating system **340** controls processing by processor **305**, and is typically stored in data storage **330** and loaded into internal storage **335** (as illustrated) for execution. The synchronization agent **130**, which cooperates with the base system **250** (FIG. **2**) for synchronizing local workspace data **245** with workspace data **135**, also may be stored in data storage **330** and loaded into internal storage **335** (as illustrated) for execution by processor **305**. The workspace data **135** may be stored in data storage **230**.

One skilled in the art will recognize that the system **100** may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system **100** in alternative ways. For example, a CRSM reader **345** such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus **310** for reading a CRSM **350** such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the system **100** may receive programs and data via the CRSM reader **345**.

FIG. **4** is a block diagram illustrating details of the remote client **120**. The client **120** includes a processor **405**, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a communications channel **410**. The client **120** further includes an input device **415** such as a keyboard and mouse, an output device **420** such as a CRT display, data storage **425** such as a magnetic disk, and internal storage **430** such as RAM, each coupled to the communications channel **410**. A communications interface

6

425 couples the communications channel **410** to the computer network **125**.

An operating system **440** controls processing by processor **405**, and is typically stored in data storage **430** and loaded into internal storage **435** (as illustrated) for execution. The PIM **170** and assistant **185** may be stored in data storage **430**, and loaded into internal storage **435** (as illustrated) for execution by the processor **405**.

One skilled in the art will recognize that the system **100** may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system **100** in alternative ways. For example, a CRSM reader **445** such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus **310** for reading a CRSM **450** such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the system **100** may receive programs and data via the CRSM reader **445**.

FIG. **5** is a block diagram illustrating a PIM interface **500**, which includes a header **505** and a selection window **510**.

The header **505** includes a synchronize button **540** and a “borrow me” button **545**, which are presented by the assistant **175**, **180** or **185** incorporated in the PIM **160**, **165** or **170**. Invoking the synchronize button **540** causes the assistant **175**, **180** or **185** to enable synchronization of data entered into the PIM **160**, **165** or **170** with the workspace data **135** on the global server **135**. The synchronize button **540** may enable the user to configure a preference file that indicates when automatic synchronization is to initiate and may also enable a user to effect manual synchronization.

The “borrow me” button **545** enables a user to use a PIM **160**, **165** or **170** for viewing and manipulating workspace data **135** downloaded from the global server **105**. That is, invoking the “borrow me” button **545** causes the corresponding assistant **175**, **180** or **185** to communicate with the global server **105**, to provide user identification and authentication information to the global server **105**, to download workspace data **135** from the global server **105**, to display and enable manipulation of the downloaded data **135** using the PIM interface **500**, and to synchronize the manipulated downloaded data **135** upon logout. Since the PIM interface **500** is provided by the pre-existing PIM, the assistant **175**, **180** or **185** need not provide its own data interface. Only a single interface is needed.

It will be appreciated that upon logout, the base systems **140** and **150** will cooperate with the synchronization agent **130** to synchronize automatically the workspace data **135** on the global server **105** with the workspace data **145** and **155**. Accordingly, the user always has access to the most updated versions of workspace data from any site that executes a PIM **160**, **165** or **170** having an assistant **175**, **180** or **185** embodied therein.

It will be appreciated that the synchronize button **540** is most helpful to the work client **110** and the home client **115**, since typically the work client **110** and home client **115** will set the preference file to configure automatic synchronization. Synchronization of the manipulated workspace data **135** at the remote client **120** will most often be effected through the automatic logout procedures of the “borrow me” button. Logout is described in greater detail with reference to the Outlook™ and Lotus Organizer examples shown and described below with reference to FIG. **7**. Accordingly, the borrow me button **545** is most helpful to the remote client **120**.

6,151,606

7

The selection window **510** provides a list of buttons **507**, wherein each button **507** corresponds to a set of workspace elements, e.g., e-mails **515**, contacts **520**, files **525**, calendar data **530** and bookmarks **535**. A mouse-down on a virtual button **507** causes the selection of a corresponding workspace element set and the selection of a corresponding user interface for displaying and enabling manipulation of the workspace elements included in the set. For example, selection of button **515** selects the e-mail set, and selects a corresponding user interface for displaying, writing, forwarding, etc. e-mails. Selecting a button **507** causes the assistant **175**, **180** or **185** to download the corresponding workspace data **135**, and causes the PIM **160**, **165** or **170** to display and enable manipulation of the downloaded data **135** on a workspace element set interface (shown and described with reference to FIG. 6).

FIG. 6 illustrates an example e-mail workspace element set user interface **600** (commonly referred to as the "In-Box") for displaying received e-mails. The user interface **600** includes a header **605**, an e-mail list window **610** and a manipulation command window **650**.

The header **605** lists the name of the workspace element set, namely, "E-Mail." The e-mail list window **610** comprises three columns, including an origin column **615** which provides the origin of each e-mail, a subject column **620** which provides the subject of each e-mail, and a date column **625** which provides the date each e-mail was received. The e-mail list window **610** may display e-mails stored in a local e-mail database (not shown), e-mails stored in the e-mail server (not shown) or e-mails downloaded from the global server **105**. The e-mails shown include a first e-mail from Joe Smith, a second e-mail from Tom Jones, and a third e-mail from Roy White. If the user depressed the "borrow me" button **545** shown in FIG. 5, then the e-mail list displayed would be the e-mails stored and downloaded from the global server **105**.

The manipulation window **650** includes available functions such as the conventional e-mail read function **630**, e-mail reply function **635**, e-mail forward function **640** and new e-mail write function **645**. It will be appreciated that the columns and functions will vary based on the PIM.

FIG. 7 is a block diagram illustrating details of a generic assistant **700**, which exemplifies each of the assistant **175**, **180** and **185**. The generic assistant **700** includes a communications module **705**, locator modules **710**, a general synchronization module **715**, a content-based synchronization module **720**, a security module **725**, an instantiator **730**, a data reader **735**, a PIM Application Program Interface (API) **740** and a de-instantiator **745**. The synchronization function of the assistant **700** uses the communications module **705**, the locator modules **710**, the general synchronization module **715**, the content-based synchronization module **720**, the security module **725** and the PIM API **740**. The "borrow me" function of the assistant **700** uses the communications module **705**, the locator modules **710**, the security module **725**, the instantiator **730**, the data reader **735**, the PIM API **740** and the de-instantiator **745**.

The communications module **705** includes routines for compressing and decompressing data, and routines for communicating with the synchronization agent **130**. The communications module **705** may apply Secure Socket Layer (SSL) technology to establish a secure communication channel. Examples of communications modules **705** may include TCP/IP stacks or the AppleTalk protocol.

The locator modules **710** include routines for identifying the memory locations of the workspace elements in the

8

workspace data **135**. Workspace element memory location identification may be implemented using intelligent software, i.e., preset memory addresses or the system's registry, or using dialogue boxes to query the user. Accordingly, the locator modules **710** determine the memory addresses of the workspace elements in e-mail workspace data **135**, in file workspace data **135**, in calendar workspace data **135**, etc.

The general synchronization module **715** examines the workspace data **135** on the global server **105** to determine whether it had been modified while the user manipulated the data on the client **110**, **115** or **120**. Further, the general synchronization module **715** determines whether the user manipulated any data on the client **110**, **115** or **120**. If the general synchronization module **715** determines that only the data on the client **110**, **115** or **120** was manipulated, then the general synchronization module **715** computes and sends the changes to the synchronization agent **130** of the global server **105**. The general synchronization module **715** is initiated when the synchronization button **540** is depressed and during the logout procedures of the "borrow me" function.

The synchronization agent **130** then updates a last synchronization signature to indicate to all base systems **140** and **150** that synchronization with workspace data **145** and synchronization with workspace data **155** are needed. If the general synchronization module **715** determines that changes were made only to the workspace data **135** on the global server **105**, then the general synchronization module **715** instructs the synchronization agent **130** to compute and transmit the changes made to the client **110**, **115** or **120** at the client's request. The client **110** or **120** then updates its information. It will be appreciated that sending only the changes reduces processor load and increases transmission line efficiency, although alternatively an entire manipulated workspace element can be sent to the global server **105**.

If the general synchronization module **715** determines that the workspace data **135** on the global server **105** has been modified since download, and that the data on the client **110**, **115** or **120** has been modified, then the general synchronization module **715** instructs the content-based synchronization module **720** to perform its duties. The content-based synchronization module **720** includes routines for reconciling two or more modified versions of a workspace element. The content-based synchronization module **720** may request a user to select the preferred one of the modified versions or may respond based on preset preferences, i.e., by storing both versions in both stores or by integrating the changes into a single preferred version which replaces each modified version at both stores.

The security module **725** includes routines for obtaining user identification and authentication using such techniques as obtaining login and password information, obtaining a response to a challenge, obtaining a public key certificate, etc. The security module **725** performs identification and authentication techniques to confirm authorization by the user to access the workspace data **135** stored on the global server **105**. It will be appreciated that authorization may be granted only to portion of the workspace data **135** that belongs to the user.

The instantiator **730** is an application program interface **730** that creates a window for displaying and enabling manipulation of the workspace data **135** downloaded from the global server **105**. In an object-oriented environment, the instantiator **730** may create a new instance for the workspace data **135**. Alternatively, the instantiator **730** may store the

local data to a buffer (not shown) and use the current interface to display and enable manipulation of the workspace data 135.

The data reader 735 communicates with the synchronization agent 130 at the global server 105, and retrieves the workspace data 135 requested. For example, if the user depresses the “borrow me” button 545 (FIG. 5) and depresses the e-mail button 515, then the data reader 735 retrieves the e-mail workspace elements of the workspace data 135, and delivers them to the PIM API 740.

The PIM API 740 translates and transfers the workspace data 135 received from the global server 105 to the PIM 160, 165 or 170 for display and enabling manipulation thereto. The PIM API 740 further translates and transfers the workspace data manipulated on the client 110, 115 or 120 from the PIM 160, 165 or 170 back to the global server 105.

The de-instantiator 745 returns the PIM 160, 165 or 170 to the state before the user selected the “borrow me” button 545. The user may initiate operations of the de-instantiator 745 by depressing an “unborrow me” button (not shown) that is presented after selection of the “borrow me” button 545. The de-instantiator 745 deletes any instance created by the instantiator 730, deletes all workspace data 135 and data created by the user on the client 110, 115 or 120 and automatically initiated synchronization of any manipulated downloaded data 135 with the workspace data 135 stored at the global server 105.

Operations of the instantiator 730, the data reader 735, the PIM API 740 and the de-instantiator 745 are described in greater detail with reference to the following examples:

OUTLOOK EXAMPLE

Action	Global Data	Local Data
standby	—	local data → pst ^{local}
button depressed	—	pst ^{local}
enter login/ password	—	pst ^{local}
authenticate	—	pst ^{local}
send global data	global data → pst ^{local}	local data → pst ^{buffer}
manipulate data	global data → global data 2	pst ^{buffer}
logout	1) Compute Δglobal data 2) Synchronize Δglobal data with global server 3) Delete global data 2	local data → pst ^{local}
	4)	

As illustrated by the Outlook™ example above, during standby, the PIM 160, 165 or 170 stores the local data on the client 110, 115 or 120 in a personal folder store pst^{local}. The user then depresses the “borrow me” button 545. The security module 725 requests the user to enter a login and password, which the global server 105 authenticates. During these steps, it will be appreciated that the local data remains stored in pst^{local}. Upon user identification and authentication, the global server 105 sends the workspace data 135 (global data) to the requesting client 110, 115 or 120. The instantiator 730 on the client 110, 115 or 120 transfers the local data from pst^{local} to a buffer pst^{buffer}, and stores the received global data into pst^{local}. The data reader 745 and PIM API 740 enable the user to manipulate the global data, the manipulated data being referred to herein as “global data 2.” Upon logout, for example, after an “unborrow me” button (not shown) is depressed, the global data 2 is synchronized with the workspace data 135. Namely, the general synchronization module 715 determines the changes

made (Δglobal data), and synchronizes Δglobal data with the workspace data 135. The de-instantiator 745 deletes local data 2 and Δglobal data, and returns the local data to pst^{local}.

LOTUS ORGANIZER EXAMPLE

Action	Global Data	Local Data
standby	—	local.org
button	—	local.org
enter login/password	—	local.org
authenticate	new instance	local.org
send global data	open with global.org	local.org
manipulate data	global.org → global.org ²	local.org
logout	1) compute Δglobal.org 2) Synchronize Δglobal.org with global server 3) delete global.org ²	local.org

As illustrated by the Lotus Organizer example above, during standby, the PIM 160, 165 or 170 stores the local data on the client 110, 115 or 120 in local.org. The user then depresses the “borrow me” button 545. The security module 725 requests the user to enter a login and password, which the global server 105 authenticates. During these steps, it will be appreciated that the local data remains stored in local.org. Upon user identification and authentication, the global server 105 sends the workspace data 135 (global data) to the requesting client 110, 115 or 120. The instantiator 730 on the client 110, 115 or 120 creates a new instance, e.g., a new window, of PIM API 740 and stores the received global data into another file, i.e., global.org. The data reader 745 and PIM API 740 enable the user to manipulate the global data, the manipulated data being referred to herein as “global data 2.” Upon logout, the global data 2 is synchronized with the workspace data 135. Namely, the general synchronization module 715 determines the changes made (Δglobal data), and synchronizes Δglobal data with the workspace data 135. The de-instantiator 745 deletes global data 2, Δglobal data and global.org.

FIG. 8 is a flowchart illustrating a method 800 of accessing data remotely in accordance with the present invention. The method 800 begins with the processor 405 in step 805 opening the PIM 160, 165 or 170 per user request, and the PIM 160, 165 or 170 opening a PIM interface 500 (FIG. 5). The PIM 160, 165 or 170 in step 810 receives a “borrow me” request from the user, i.e., the user depresses the “borrow me” button 545. The PIM API 740 in step 815 recognizes the request, and instructs the communications module 705 to create a communications link with the global server 105.

The security module 725 in step 820 requests and transmits identification and authentication information such as login and password information from the user to the global server 105 for examination. If the global server 105 fails to identify or authenticate the user, then the method 800 ends. Otherwise, the instantiator 730 in step 825 opens a PIM interface 500 to display and enable manipulation of the workspace data 135 downloaded from the global server 105. The data reader 735 in step 830 reads the workspace data 135 downloaded from the global server 105, and in step 835 translates the data to the appropriate format if necessary. That is, the data reader 735 translates the workspace data 135 from the format implemented by the global server 105 to the format implemented by the PIM 160, 165 or 170. The PIM API 740 in step 840 passes the translated workspace data 135 to the PIM interfaces 500 and 600.

The PIM 160, 165 or 170 enables the user in step 845 to manipulate the workspace data 135 as necessary. Manipu-

6,151,606

11

lation includes adding new data, deleting workspace data **135**, editing workspace data **135**, etc. For example, the user can depress the e-mail button **515** in interface **500** to select, review and manipulate e-mail in interface **600**, and then can depress the calendar button **530** in interface **500** to select, review and manipulate calendar information (not shown) in an interface similar to the e-mail interface **600**. In step **850**, the PIM API **740** waits to receive an "end session" request. Until an "end session" request is received, the method **800** returns to step **830** to enable continued data review and manipulation.

Upon receiving an "end session" or "unborrow me" request, the de-instantiator **745** initiates the general synchronization module **715** in step **855** to synchronize the manipulated workspace data on the client **110**, **115** or **120** with the workspace data **135** on the global server **105**, if required. Synchronization is described in greater detail with reference to FIG. **9**. The de-instantiator **745** in step **860** deletes the workspace data on the client **110**, **115** or **120**, and deletes all records of the matter. Method **800** then ends.

FIG. **9** is a flowchart illustrating a method **900** for synchronizing workspace data in a computer network **100**. Method **900** begins with the communications module **705** in step **905** establishing a communications link with the synchronization agent **130** of the global server **105**. The locator modules **710** in step **910** identify the memory locations of the workspace elements in the workspace data **135**. It will be appreciated that workspace element memory location identification may be implemented using intelligent software or dialogue boxes.

The general synchronization module **715** in step **915** compares version information (not shown) for each workspace element in the workspace data (on the client **110**, **115** or **120** and on the global server **105**) against a last synchronization signature to determine which workspace elements have been modified. In this embodiment, a workspace element may have been modified if the date and time of the last modification is after the date and time of the downloading.

If the general synchronization module **715** locates no modified workspace elements in the workspace data on the client **110**, **115** or **120**, then the method **900** ends. Otherwise, the general synchronization module in step **920** determines whether the version of the same workspace element of the workspace data **135** on the global server **105** has been modified since the data **135** was downloaded.

If only the version on the client **110**, **115** or **120** has been modified, then the general synchronization module **715** in step **925** stores the updated version of the workspace element at the global server **105**. To store the updated version on the global server **105**, the general synchronization module **715** may compute the changes made and forward the changes to the synchronization agent **130**. The synchronization agent **130** enters the changes into the global server **105** version. The general synchronization module **715** in step **930** determines whether all workspace elements downloaded to the client **110**, **115** or **120** have been examined. If not, then method **900** returns to step **915**. Otherwise, the synchronization agent **130** in step **935** updates the last synchronization signature, and method **900** ends. Updating the last synchronization signature will instruct the base systems **140** and **150** to synchronize the workspace data **145** and **155** with the workspace data **135** on the global server **105**, as described in the patent applications incorporated by reference above.

If the general synchronization module **715** in step **920** determines that both the version on the client **110**, **115** or **120**

12

and the version on the global server **105** have been modified, then the general synchronization module in step **935** instructs the content-based synchronization module **729** to reconcile the modified versions. Reconciliation may include requesting instructions from the user, or performing based on pre-selected preferences responsive actions such as storing both versions at the global server **105**. The general synchronization module **715** in step **940** stores the preferred version on the global server **105**. Method **900** then proceeds to step **930**.

The foregoing description of the preferred embodiments of the present invention is by way of example only, and other variations and modifications of the above-described embodiments and methods are possible in light of the foregoing teaching. Although the network sites are being described as separate and distinct sites, one skilled in the art will recognize that these sites may be a part of an integral site, may each include portions of multiple sites, or may include combinations of single and multiple sites. Further, components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. Connections may be wired, wireless, modem, etc. The embodiments described herein are not intended to be exhaustive or limiting. The present invention is limited only by the following claims.

What is claimed is:

1. A computer-based method, comprising the steps of:

executing a workspace data manager on an untrusted client site;

requesting the workspace data manager to access data temporarily from a remote site, the remote being connected via a network to untrusted client site;

initiating a communications channel with the remote site;

downloading data from the remote site;

placing the data in temporary storage on the untrusted client site;

using the workspace data manager to present the downloaded data; and

automatically disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data.

2. The method of claim 1, further comprising the step of requesting the workspace data manager to provide an interface for enabling presentation of the downloaded data.

3. The method of claim 2, further comprising the steps of using the workspace data manager to manipulate the downloaded data, thereby creating manipulated data, using the workspace data manager interface to request synchronization, and synchronizing the manipulated data with the data at the remote site.

4. The method of claim 3, wherein the data at the remote site has not been modified after the step of downloading and before the step of synchronizing and therefore includes the downloaded data.

5. The method of claim 3, wherein the data at the remote site has been modified after the step of downloading and before the step of synchronizing, and therefore is different than the downloaded data.

6. The method of claim 2, wherein the workspace data manager provides an interface by creating an instance.

7. The method of claim 2, wherein the workspace data manager provides an interface by providing access to its only interface.

8. The method of claim 1, further comprising the step of translating the downloaded data from the format used by the remote site and the format used by the workspace data manager.

6,151,606

13

9. The method of claim 1, further comprising the step of deleting the workspace data manager interface after it is no longer needed.

10. A system on an untrusted client site, comprising:

a communications module for download data from a remote site, the remote site being connected via a network to the untrusted client site;

program code for placing the downloaded data in temporary storage on the untrusted client site;

an application program interface coupled to the communications module for communicating with a workspace data manager to present the downloaded data; and

program code coupled to the application program interface for automatically disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data.

11. The system of claim 10, further comprising an instantiator for requesting the workspace data manager to provide an interface for enabling presentation of the downloaded data.

12. The system of claim 11, wherein the workspace manager enables manipulation of the downloaded data to create manipulated data and the data manipulation interface enables a request to synchronize the data, and further comprising a synchronization module coupled to the communications module for enabling synchronization of the manipulated data with the data at the remote site.

13. The system of claim 12, wherein the data stored at the remote site has not been modified and therefore includes the downloaded data.

14. The system of claim 12, wherein the data stored at the remote site has been modified, and therefore is different than the downloaded data.

15. The system of claim 14, further comprising a content-based synchronization module for synchronizing the data stored at the remote site with the manipulated data.

16. The system of claim 11, wherein the workspace data manager creates another instance of the interface to enable presentation of the downloaded data.

17. The system of claim 11, wherein the workspace data manager provides access to its only interface to enable presentation of the downloaded data.

14

18. The system of claim 11, further comprising a deinstantiator for deleting the interface after it is no longer required.

19. The system of claim 10, further comprising a data reader for translating the downloaded workspace data from the format used by the remote site to the format used by the workspace data manager.

20. A system comprising:

means for executing a workspace data manager on an untrusted client site;

means for requesting the workspace data manager to access data temporarily from a remote site, the remote site being connected via a network to the untrusted client site;

means for initiating a communications channel with the remote site;

means for downloading data from the remote site;

means for placing the data in storage on the untrusted client site;

means for using the workspace data manager to present the downloaded data; and

means for disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data.

21. A computer-readable storage medium storing program code for causing a computer to perform the steps of:

executing a workspace data manager on an untrusted client site;

requesting the workspace data manager to access data temporarily from a remote site, the remote site being connected via a network to the untrusted client site;

initiating a communications channel with the remote site;

downloading data from the remote site;

placing the data in temporary storage on the untrusted client site;

using the workspace data manager to present the downloaded data; and

automatically disabling the untrusted client site from accessing at least a portion of the downloaded data after a user has finished using the data.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 6,151,606

DATED : November 21, 2000

INVENTOR(S) : Daniel J. Mendez

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Column 12, line 47, after "manipulate the" change the word "dow" to -- downloaded --

Column 13, line 5, after "module for" change the word "download" to -- downloading --

Column 13, line 23, before "manager" insert -- data --

Column 14, line 18, after "placing the data in" insert -- temporary --

Signed and Sealed this

Eighth Day of May, 2001



Attest:

NICHOLAS P. GODICI

Attesting Officer

Acting Director of the United States Patent and Trademark Office